12-07-05

AF

TRANSMITTAL OF APPEAL BRIEF

Docket No.
06727/000H560-US0

In re Application of: Daniel Geist et al.

Application No.	Filing Date	Examiner	Group Art Unit
09/605,334-Conf. #3652	June 27, 2000	T. H. Stevens	2123

Invention: IDENTIFICATION OF MISSING PROPERTIES IN MODEL CHECKING

TO THE COMMISSIONER OF PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed: October 4, 2005 .

The fee for filing this Appeal Brief is $ 500.00 .

[x] Large Entity [ ] Small Entity

[ ] A petition for extension of time is also enclosed.

The fee for the extension of time is _____ .

[x] A check in the amount of $ 500.00 is enclosed.

[ ] Charge the amount of the fee to Deposit Account No. _____ .
This sheet is submitted in duplicate.

[ ] Payment by credit card. Form PTO-2038 is attached.

[x] The Director is hereby authorized to charge any additional fees that may be required or credit any overpayment to Deposit Account No. 04-0100 .
This sheet is submitted in duplicate.

Dated: December 2, 2005

S. Peter Ludwig
Attorney Reg. No. : 25,351
DARBY & DARBY P.C.
P.O. Box 5257
New York, New York 10150-5257
(212) 527-7770

Express Mail Label No. Dated: _____

Serial No. 09/605,334

Customer No. 07278 Attorney Docket No. 06727/000H560-US0

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	: Geist et al.	
Serial No.	: 09/605,334	Group Art Unit: 3652
Filed	: June 27, 2000	Examiner: Thomas Stevens
For	: IDENTIFICATION OF MISSING PROPERTIES IN MODEL CHECKING	

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

APPEAL BRIEF

(1) Real Party in Interest

The subject application is owned jointly by International Business Machines Corporation, having a place of business at New Orchard Road, Armonk, New York, and by Marvell Semiconductor Israel Ltd., having a place of business at 6 Hamada Street, Yokneam, Israel. The assignments were recorded in the U.S.P.T.O. on October 26, 2000, under Reel 011206, Frame 0073, and on July 6, 2005, under Reel 016474, Frame 0996.

(2) Related Appeals and Interferences

None.

(3) Status of Claims

This application as filed included claims 1-32. The claims have not been amended. All the claims were finally rejected in an Official Action dated May 4, 2005.

On October 3, 2005, Appellant appealed from the rejection of claims 1-32 (all the claims currently pending in this application).

(4) Status of Amendments

No amendments have been made.

(5) Summary of Claimed Subject Matter

Appellant's invention, as recited in independent claims 1, 19 and 21, provides a method, processor and computer software product for use in verification of a system by model checking. The invention is directed specifically to evaluating how well a specification of a set of properties that is to be used in model checking covers the model that it is supposed to check.

In model checking, a verification engineer writes a set of properties (known as a *specification*) that a design of a target system is expected to fulfill. An *implementation model* is then tested to ascertain that the model satisfies all of the properties in the set (page 1, lines 14-28, in the present patent application). Before the present invention, however, the verification engineer had no systematic way to be sure of whether the property set is complete (page 2, lines 14-18). If the property set is incomplete, a bug in the design may go undetected.

The method recited in claim 1 comprises the following steps:

(a) Providing an implementation model, which defines model states of a target system and model transitions between the model states.

(b) Providing a specification of the target system, comprising properties that the system is expected to obey.

(c) Creating a tableau from the specification. A "*tableau*" is defined explicitly in the present patent application as "a finite state machine that satisfies all of the properties in the specification" (page 4, lines 14-16). Thus, as recited in the claim, the tableau comprises "tableau states" with "tableau transitions" between the states in accordance with the properties in the specification.

(d) Comparing the tableau transitions to the model transitions to determine whether a discrepancy exists therebetween.

In other words, the method of claim 1 creates and compares the transitions of two entities: the implementation model, which expresses the design, and the tableau, which expresses the specification. Fig. 2 in the present patent application shows the model of a target hardware device (40) and the tableau (50) that are created in the course of testing the design of the device (in this case a two-port arbiter – page 12, lines 16-20). The tableau can be seen as "a sort of virtual device corresponding to the actual target device" (page 17, lines 5-6). "To the extent that there are no substantive differences" between the transitions of the model and the tableau, "it is concluded that the set of properties fully specifies the model" (page 4, lines 20-22).

The steps of the method of claim 1 are shown, in two different embodiments, in Figs. 4 and 5 of the present patent application and are described with reference thereto in the specification. Step (a), providing an implementation model, is implicit in these figures and is shown explicitly in the example shown in Table I (page 13). Step (b), providing a specification, is described on page 17, lines 24-28, and is shown by way of example on page 14, lines 3-27, and in symbolic form in Table II (page 15). Step (c), creating a tableau, is described on page 17, lines 28-32. Fig. 3 (in conjunction with the description on page 16, line 8 – page 17, line 4) shows the complete tableau corresponding to the specification in Table II. Finally, step (d), comparing the tableau and model transitions, is described generally on page 17, lines 8-18. The remaining stages in Figs. 4 and 5 present detailed realizations of two methods for carrying out the comparison.

Claim 19 recites a verification processor that carries out the steps of claim 1, while claim 21 recites software that causes a computer to perform these functions. The processors and software are described in the specification on page 26, lines 1-12. The processor (22) is shown in Fig. 1.

Independent claims 24, 31 and 32 define a method, apparatus and computer software product according to another aspect of the present invention. The method

recited in claim 24 comprises the same first three steps as claim 1, but concludes with a different final step:

(d) Comparing the model and the tableau by inputting the model and the tableau to an automatic model checking program.

This step is shown explicitly in the "MODEL CHECK" step of Fig. 4, and is described on page 18, line17 – page 19, line 5.

Claim 31 recites model checking apparatus that carries out the steps of claim 24, while claim 32 recites software that causes a computer to perform these functions. As noted above, the apparatus is shown in Fig. 1, and both the processor and software are described in the specification on page 26, lines 1-12.

The remaining claims each depend from one of the above-mentioned independent claims.

(6) Grounds of Rejection to be Reviewed on Appeal

All the claims currently pending in this application were rejected under 35 U.S.C. 103(a) over Cleaveland ("Tableau-Based Model Checking...") in view of Cleaveland et al. (U.S. Patent 6,385,765). Appellant believes this rejection should be reversed

(7) Argument

I. The Section 103(a) Rejection of Claims 1, 19 and 21

Appellant respectfully submits that the Examiner erred in maintaining that it would have been obvious to a person of ordinary skill in the art to combine the teachings of Cleaveland and Cleaveland et al. so as to arrive at the inventive method, processor and software product recited by claims 1, 19 and 21.

Claims 1, 19 and 21 each recite creating a tableau from a specification. The tableau defines tableau states with tableau transitions between the tableau states. The implementation model similarly has model states and model transitions. According to these claims, the model transitions are compared to the tableau transitions in order to determine whether a discrepancy exists between the two.

MPEP 2143.03 states:

> To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). "All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Applying this statement to the present case, the question to be answered is whether the cited art teaches or suggests providing or creating two entities - a model and a tableau - having respective states and transitions, and then comparing the transitions of the two entities to determine whether there is a discrepancy. If not, then the Examiner has failed to make a *prima facie* case of obviousness.

Cleaveland describes a procedure for model checking using a type of logic known as "mu-calculus." As noted in the abstract, Cleaveland uses a "tableau-based proof system" to prove that this type of logic is "sound and complete," i.e., that the mu-calculus itself is suitable for use in model checking. He gives an example of a "tableau rules," consisting of rules R1-R8, in Figure 3 on page 6. He then defines a tableau as follows (page 7, lines 9-10): "a tableau for σ is a maximal proof tree having σ as its root and constructed using R1-R8," and shows a sample tableau in Figure 5 on page 9.

Cleaveland's "tableau," in other words, is a hierarchy of rules. Despite the chance semantic similarity to the term used in the present patent application, Cleaveland's tableau (as can be seen clearly in his Figure 5) has no tableau states and no tableau transitions. The only states and transitions that Cleaveland does discuss (page 3, line 21 – page 4, line 2) are those of the model *M* itself, not of the tableau. The fact that Cleaveland's proof system is "tableau-based" simply means that "proofs are conducted in a top-down fashion" (page 6, lines 10-12). Cleaveland neither teaches nor suggests that a tableau could define tableau states with tableau transitions, as required by claims 1, 19 and 21.

Cleaveland et al. describes a method for specifying concurrent systems (abstract). One aspect of the method involves model checking, which "allows one to check whether a state-machine representation of a system is a model of, or satisfies, a

formula in temporal logic. Temporal logic allows for the specification of critical system properties..." (col. 2, lines 30-37). This is the traditional definition of model checking, which is also explained in the Background of the present patent application (page 1, lines 14-28). Cleaveland et al. speaks of the "state machine" in the conventional way: as the model of the design itself, which is tested against the rules of the specification. Cleaveland et al. makes no mention or suggestion of creating a tableau from the specification, whether in the sense of a state machine, as recited in claim 1, or a maximal proof tree as defined by Cleaveland. Clearly, therefore, Cleaveland et al. cannot be taken to suggest creating a tableau with tableau states and tableau transitions.

Thus, to summarize, neither Cleaveland nor Cleaveland et al. makes any suggestion of comparing tableau transitions to model transitions, as required explicitly by claims 1, 19 and 21. Therefore, these claims are plainly patentable over the cited art.

In rejecting claim 1, the Examiner stated that Cleaveland et al. (referred to by the Examiner as "Patent") teaches creating a tableau from the specification, and that Cleaveland (referred to by the Examiner as "Paper") teaches defining tableau states with tableau transitions. The passage cited by the Examiner in Cleaveland et al. (col. 2, lines 30-37) in support of this contention is simply a background definition of model checking, and makes no reference to or suggestion of a tableau, whether as a state machine based on the specification or a maximal proof tree. The cited passage in Cleaveland (section 3, particularly page 7, second paragraph) relates to a tableau, but does not even hint that the tableau might have states or transitions. The Examiner then referred back to the same passage in Cleaveland et al. (col. 2, lines 30-37) as purportedly teaching the step of comparing tableau transitions to model transitions. The cited passage, however, makes no mention of any sort of transitions, let alone comparing the transitions of the model and the tableau as required by claim 1.

In the Examiner's response to Appellant's arguments in the present Official Action (paragraph 5, page 2), the Examiner stated that Cleaveland et al. notes that mu-calculus is used to eliminate redundant portions of the state space (citing col. 4, lines 35-40), and that "Without creating a tableau via mu-calculus... the simulation of the state machine for redundant states would not occur..." This statement is incorrect.

Cleaveland et al. states that the disclosed model checker uses a "new kind of partial order reduction method to eliminate redundant portions of the state space from analysis" (col. 4, lines 32-34). This technique is "targeted for use with the modal mu-calculus" (col. 4, lines 34-36). In other words, Cleaveland et al. uses partial order reduction to eliminate redundant states and transitions of the model, in a way that is said to make the model well suited for use with mu-calculus. Even if the Examiner's statement that mu-calculus is used to eliminate redundant state space were correct, however, it still does not establish that Cleaveland et al. provides and compares tableau transitions to model transitions, as required by claims 1, 19 and 21.

The Examiner then went on to note (page 4 of the Official Action) that Appellant's statement that "in model checking a verification engineer writes a set of properties... that a design is expected to fulfill," is equivalent to certain references to model checking in Cleaveland et al. (col. 3, lines 16-19 and 65-66). Appellant does not disagree with this finding. The statement in question, however, is simply a general, background definition of model checking. Appellant does not claim to have invented model checking, but rather a method, processor and software for use in the context of model checking based on comparing tableau transitions to model transitions. The Examiner has failed to show any suggestion of this invention in the cited art.

II. The Section 103(a) Rejection of Claims 25, 31 and 32

Appellant respectfully submits that the Examiner erred in maintaining that it would have been obvious to a person of ordinary skill in the art to combine the teachings of Cleaveland and Cleaveland et al. so as to arrive at the inventive method, processor and software product recited by claims 25, 31 and 32.

Claims 25, 31 and 32, like claims 1, 19 and 21, provide an implementation model and a tableau, both of which have states and transitions. They are compared by inputting both of them to an automatic model checking program. The Examiner rejected claims 25, 31 and 32 on grounds that were similar to the grounds of rejection of claims 1, 19 and 21, with the additional citation of col. 4, lines 10-19, in Cleaveland et al. This

latter passage is alleged to teach the step of comparing the model and the tableau by inputting them to an automatic model checking program.

As explained above in reference to claims 1, 19 and 21, neither Cleaveland nor Cleaveland et al. teaches or suggests the step of creating a tableau having tableau states and tableau transitions, as defined in the present patent application. In the cited references, there is only a single set of states and transitions: those of the model itself, as in methods of model checking known in the art. Furthermore, there is no suggestion in either Cleaveland or Cleaveland et al. that any sort of tableau might be input to an automatic model checking program. The passage cited by the Examiner in this regard (col. 4, lines 10-19) refers only to the use of an automatic model checker in general, and not to the specific use of a model checker for comparing a model and a tableau, as recited in claims 25, 31 and 32.

Thus, neither Cleaveland nor Cleaveland et al. makes any suggestion of providing a model and creating a tableau, both having respective states and transitions and comparing these two entities by inputting them to an automatic model checking program, as required explicitly by claims 25, 31 and 32. Therefore, these claims are patentable over the cited art.

III. The Section 103(a) Rejection of Claims 2 and 26

Appellant respectfully submits that even if independent claims 1 and 25 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claims 2 and 26.

Claims 2 and 26 state that the tableau is created by defining a finite state machine using a hardware description language. Hardware description languages, such as VHDL or Verilog, are used in the design of electronic circuits and are commonly used in creating implementation models for model checking (page 1, lines 23-28, in the present patent application). Specification properties, however, are rules, which are generally written in a temporal language (page 1, lines 3-12, in the present patent

application, as well as Cleaveland et al., col. 2, lines 33-37). Claims 2 and 26 recite a novel approach wherein the specification can be used to create a sort of "virtual device" in hardware description language, which can then be conveniently compared to the actual device under test (page 17, lines 5-19, in the present patent application).

In rejecting claims 2 and 26, the Examiner asserted that the added limitation that these claims recite is taught by Cleaveland et al. (col. 3, lines 7-10). The cited passage refers to VPL, or "value passing language," which is a "textual specification language used to define system components that exchange data values and that exhibit complex control structure." VPL is used to "create networks and processes" (col. 2, lines 64-65). Sample VPL code is shown by Cleaveland et al. in Fig. 2. Cleaveland et al. do not provide sufficient information regarding VPL to enable a person of ordinary skill in the art to use it for any purpose. Based on the very limited description provided by Cleaveland et al., however, it is clear that VPL is not a "hardware description language." Furthermore, even if VPL were considered to be a hardware description language, there is no suggestion in Cleaveland et al. that VPL could be used to create a tableau by defining a finite state machine.

Thus, Appellant respectfully submits that claims 2 and 26 are independently patentable over the cited art.

IV. The Section 103(a) Rejection of Claims 3 and 27

Appellant respectfully submits that even if claims 2 and 26 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claims 3 and 27.

Claims 3 and 27 state that the tableau is described as a virtual device, which has inputs and outputs corresponding to model inputs and outputs of the implementation model. As noted earlier, this approach is illustrated in Fig. 2 of the present patent application.

In rejecting claims 3 and 27, the Examiner equated the "virtual device" recited in the claims to "the device to be tested." This analysis of the claim is wrong on its face,

since it relates to the virtual device and to the actual device represented by the implementation model as though they were one and the same. Claims 3 and 27 explicitly refer to two entities having corresponding inputs and outputs, but the Examiner appears to have ignored this limitation. The Examiner went on to cite col. 3, lines 62-67, in Cleaveland et al., which mentions "equivalence checking," but Cleaveland et al. fails to give any further detail of this technique. This vague reference to a general class of verification techniques certainly does not suggest the step of "describing a virtual device having inputs and outputs corresponding to model inputs and outputs of the implementation model," as required by claims 3 and 27.

Thus, Appellant respectfully submits that claims 3 and 27 are independently patentable over the cited art.

V. The Section 103(a) Rejection of Claims 4, 22 and 28

Appellant respectfully submits that even if claims 3, 21 and 27 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claims 4, 22 and 28.

Claims 4 and 22 state that the comparison between the tableau transitions and the model transitions is carried out by performing a reachability analysis using both the implementation model and the tableau. One way of carrying out a reachability analysis is by model checking (page 1, lines 28-29, in the present patent application), which is recited in claim 28. All of claims 4, 22 and 28 state that identical inputs are provided to the implementation model and the tableau. It is then verified that the outputs of the implementation model and the tableau are also identical.

Neither Cleaveland nor Cleaveland et al. teaches or suggests the use of reachability analysis or model checking to provide identical inputs to two entities and verify that the outputs of the two entities are identical, as required by claims 4, 22 and 28. In rejecting claims 4 and 22, the Examiner alleged that Cleaveland teaches these limitations on page 19, section 5, and in Figure 7. This section and figure in Cleaveland provide a basic algorithm for "determining whether or not a state... satisfies a mu-

calculus proposition" (page 19, lines 3-4). With respect to claim 28, the Examiner cited col. 2, lines 30-46, in Cleaveland et al., and page 7, section 3, second paragraph, in Cleaveland, which respectively provide general definitions of model checking and of Cleaveland's tableau. None of these passages makes any suggestion of providing identical inputs and verifying corresponding identical outputs of two entities, whether by reachability analysis or by any other technique.

Thus, Appellant respectfully submits that claims 4, 22 and 28 are independently patentable over the cited art.

VI. The Section 103(a) Rejection of Claims 5 and 23

Appellant respectfully submits that even if claims 4 and 22 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claims 5 and 23.

Claims 5 and 23 state that the model and the tableau are compared automatically using a model checker. This limitation is similar to the limitation of claims 25 and 32, which was explained above in section II. As Appellant pointed out in section II, neither Cleaveland nor Cleaveland et al. makes any suggestion of comparing the transitions of a model and a tableau automatically using a model checking program, as required explicitly by claims 5 and 23. In the cited references, model checking is applied only to the model itself, as in methods of model checking known in the art, and there is no suggestion that any sort of tableau might be input to an automatic model checking program.

Thus, Appellant respectfully submits that claims 5 and 23 are independently patentable over the cited art.

VII. The Section 103(a) Rejection of Claim 7

Appellant respectfully submits that even if claim 1 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of

these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claim 7.

Claim 7 states that the tableau transitions are compared to the model transitions by associating model transitions with corresponding tableau transitions. In rejecting this claim, the Examiner maintained that the term "associating" is unclear and was therefore not addressed in the examination of claim 7. The Examiner raised this issue in rejecting claims 7-9 and 12 under 35 U.S.C. 112, second paragraph, in the first Official Action in this case, dated December 29, 2004. Appellant traversed the rejection, and the Examiner accordingly withdrew the rejection under 35 U.S.C. 112 in the present Official Action (see paragraph 4, page 2).

Although the Examiner acknowledged that the limitations of claims 7-9 and 12 are clear, however, he failed to cite any specific grounds to support the contention that the limitation of claim 7 is obvious over the cited art. The Examiner cited only the general references to verification, state machines, model checking, and maximal proof trees that he cited earlier in regard to the independent claims (Cleaveland et al., col. 1, lines 15-19; col. 5, lines 25-30; and col. 2, lines 30-37; and Cleaveland, page 7, section 3, second paragraph). In fact, neither Cleaveland nor Cleaveland et al. teaches or suggests associating model transitions with corresponding tableau transitions, whether in the cited passages or anywhere else in these references.

Thus, Appellant respectfully submits that claim 7 is independently patentable over the cited art.

VIII. The Section 103(a) Rejection of Claims 6 and 9

Appellant respectfully submits that even if claims 4 and 7 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claims 6 and 9.

Claims 6 and 9 recite the step of finding or providing evidence of a tableau transition that is not implemented in the model. With respect to claim 9, the Examiner maintained that the claim limitation is unclear, and he therefore did not cite any specific

grounds in support of the obviousness rejection. Regarding claim 6, the Examiner maintained that Cleaveland teaches providing evidence of a tableau transition that is not implemented in the model on page 19, section 5, and in Fig. 7, "with graphical user interface." The cited passage and figure, however, describe a "basic algorithm" for determining whether a state satisfies a mu-calculus proposition. They make no mention or suggestion of providing evidence of an unimplemented transition, as required by claims 6 and 9. Cleaveland also makes no mention of any sort of user interface. Although Cleaveland et al. describes a user interface, he does not suggest that it could be used to provide evidence of an unimplemented tableau transition.

Thus, Appellant respectfully submits that claims 6 and 9 are independently patentable over the cited art.

IX. The Section 103(a) Rejection of Claim 8

Appellant respectfully submits that even if claim 7 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claim 8.

Claim 8 states that the model transitions are associated with tableau transitions by defining a reachable simulation preorder relating the model and the tableau. In rejecting this claim, the Examiner again cited page 19, section 5, and Fig. 7, in Cleaveland, "with graphical user interface." The cited passage is completely unrelated to simulation preorders. In fact, both Cleaveland and Cleaveland et al. mention the use of preorders (for example, Cleaveland, page 12, Lemma 4.7; and Cleaveland et al., col. 5, lines 18-24). Neither of the references, however, either teaches or suggests that a reachable simulation preorder could be used to relate a model and an associated tableau, as required by claim 8.

Thus, Appellant respectfully submits that claim 8 is independently patentable over the cited art.

X. The Section 103(a) Rejection of Claim 11

Appellant respectfully submits that even if claim 9 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claim 11.

Claim 11 states that when an unimplemented transition is found in the tableau, it is used as the basis for deriving an indication that a transition of the target system is missing in the model. The Examiner gave no specific grounds for this rejection, other than simply repeating the same general references to verification, state machines, model checking, and maximal proof trees that he cited earlier in regard to the independent claims. In any case, neither Cleaveland nor Cleaveland et al. makes any mention or suggestion of deriving an indication that a transition of some target system is missing in a model of the system.

Thus, Appellant respectfully submits that claim 11 is independently patentable over the cited art.

XI. The Section 103(a) Rejection of Claim 12

Appellant respectfully submits that even if claim 1 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claim 12.

Claim 12 adds the step of associating model states with corresponding tableau states. The Examiner gave no specific grounds for this rejection, other than simply repeating the same general references to verification, state machines, model checking, and maximal proof trees that he cited earlier in regard to the independent claims. Neither Cleaveland nor Cleaveland et al. makes any mention or suggestion of associating model states with tableau states (or of associating any type of states with another type of states).

Thus, Appellant respectfully submits that claim 12 is independently patentable over the cited art.

XII. The Section 103(a) Rejection of Claim 13

Appellant respectfully submits that even if claim 12 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claim 13.

Claim 13 recites the step of finding a tableau state that is not implemented in the model. The Examiner gave no specific grounds for this rejection, other than simply repeating the same general references to verification, state machines, model checking, and maximal proof trees that he cited earlier in regard to the independent claims. In any case, neither Cleaveland nor Cleaveland et al. makes any mention or suggestion of finding a state in one entity that is not implemented in the states of another entity.

Thus, Appellant respectfully submits that claim 13 is independently patentable over the cited art.

XIII. The Section 103(a) Rejection of Claims 10, 14, 18 and 24

Appellant respectfully submits that even if claims 1, 9, 13 and 21 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claims 10, 14, 18 and 24.

Claims 10 and 14 recite the step of deriving an indication that the specification (from which the tableau is created) is not complete with respect to the implementation model of the target system. Claims 18 and 24 recite similar limitations, i.e., verifying that the specification is a complete (and correct) description of the implementation model. Neither Cleaveland nor Cleaveland et al. relates to the question of whether a specification is complete with respect to the model of a target system.

In rejecting this claim, the Examiner cited col. 6, lines 18-24 and 41-51, in Cleaveland et al. The first of these passages relates to state-space management methods, following which verified designs are submitted to automatic code generation. The second passage refers to simulation facilities, which include lists of enabled transitions

and activated breakpoints. Neither of these passages even has anything to do with a specification, let alone indicating that a specification is complete or incomplete.

Thus, Appellant respectfully submits that claims 10, 14, 18 and 24 are independently patentable over the cited art.

XIV. The Section 103(a) Rejection of Claim 15

Appellant respectfully submits that even if claim 13 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claim 15.

Claim 15 recites the step of deriving an indication that a state of the target system is missing in the model. In rejecting this claim, the Examiner asserted that the simulation menu bar in Cleaveland et al. indicates "status," and made reference to col. 6, lines 52-58, and col. 5, lines 49-58. The cited passage in col. 6 simply refers to the control of simulation options and has nothing to do with missing states. The cited passage in col. 5 states that the various checkers and simulators in the system of Cleaveland et al. rely on "operational semantics" to ensure that their activities are carried out "without the possibility of error due to spurious system behaviors or the accidental elimination of actual system behaviors." In other words, Cleaveland et al. says that enforcement of particular semantics in formally defining the system behavior ensures that there will be no errors. It would then appear that providing an indication of a missing state of the target system is superfluous for Cleaveland et al. (although it is impossible to know from the very brief description given in the patent how "operational semantics" might actually achieve error-free behavior). In any case, Cleaveland et al. makes no mention or suggestion of deriving an indication that a state of the target system is missing in the model, as required by claim 15.

Thus, Appellant respectfully submits that claim 15 is independently patentable over the cited art.

XV. The Section 103(a) Rejection of Claim 16

Appellant respectfully submits that even if claim 12 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claim 16.

Claim 16 recites the step of finding multiple model states corresponding to a single tableau state. In rejecting this claim, the Examiner cited col. 5, lines 38-46, in Cleaveland et al., and page 19, section 5, paragraphs 1-3, along with Fig. 7, in Cleaveland. The cited passage in Cleaveland et al. relates to hierarchically-structured networks of processes and the use of a "communications scheduler" to arbitrate requests for input and output. This passage has nothing whatsoever to do with model states or tableau states. The passage and figure in Cleaveland, as noted earlier, describe a "basic algorithm" for determining whether a state satisfies a mu-calculus proposition. It makes no mention or suggestion of finding multiple model states corresponding to a single tableau state, as required by claim 16.

Thus, Appellant respectfully submits that claim 16 is independently patentable over the cited art.

XVI. The Section 103(a) Rejection of Claims 29 and 30

Appellant respectfully submits that even if independent claim 25 were conceded to be unpatentable over Cleaveland in view of Cleaveland et al., the combined teachings of these references still would not have led a person of ordinary skill in the art to arrive at the additional limitations recited by dependent claims 29 and 30.

Claim 29 states that the step of comparing the implementation model and the tableau provides evidence of a transition or state in the tableau that is not implemented in the model. Claim 30 adds the limitation that the evidence is in the form of a counter-example.

In rejecting these claims, the Examiner cited section 3 of Cleaveland and col. 6, lines 38-51, and col. 1, lines 51-56, of Cleaveland et al. Section 3 in Cleaveland describes generally a proof system for establishing when states in a model satisfy formulas in the mu-calculus. It makes no mention of transitions or states in any sort of

tableau that are not implemented in the model, and fails to suggest any method that might be used to provide evidence of such transitions or states, as required by claim 29. The cited passage in col. 1 of Cleaveland et al. refers generally to avoiding failures and errors using certain computer-aided software engineering tools, but does not specify what types of errors might be avoided or how the avoidance is accomplished. There is no suggestion in this passage or anywhere else in Cleaveland et al. of providing evidence of missing states or transitions that occur in a model or a tableau.

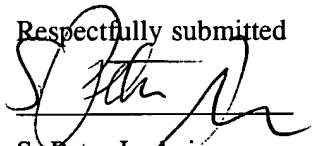
Thus, Appellant respectfully submits that claims 29 and 30 are independently patentable over the cited art.

Summary

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-32 was erroneous. Reversal of his decision is respectfully requested.

Date: December 2, 2005

Respectfully submitted



S. Peter Ludwig
Reg. No. 25,351
Attorney for Applicants

DARBY & DARBY, P.C.
P.O. Box 5257
New York, NY 10150-5257
212.527.7700

APPENDIX A

Claims pending as of the present date:

1. A method for verification, comprising:

providing an implementation model, which defines model states of a target system and model transitions between the model states;

providing a specification of the target system, comprising properties that the system is expected to obey;

creating a tableau from the specification, the tableau defining tableau states with tableau transitions between the tableau states in accordance with the properties; and

comparing the tableau transitions to the model transitions to determine whether a discrepancy exists therebetween.

2. A method according to claim 1, wherein creating the tableau comprises defining a finite state machine using a hardware description language.

3. A method according to claim 2, wherein the implementation model has model inputs and outputs, and wherein defining the finite state machine comprises describing a virtual device having inputs and outputs corresponding to the model inputs and outputs of the implementation model.

4. A method according to claim 3, wherein comparing the transitions comprises performing a reachability analysis using both the implementation model and the tableau while providing identical inputs to the inputs of both the implementation model and the tableau, and verifying that the outputs are always identical.

5. A method according to claim 4, wherein performing the reachability analysis comprises comparing the model and the tableau automatically using a model checker.

6. A method according to claim 4, wherein performing the reachability analysis comprises providing evidence of a tableau transition that is not implemented in the model.

7. A method according to claim 1, wherein comparing the tableau transitions comprises associating model transitions with corresponding tableau transitions.

8. A method according to claim 7, wherein associating the transitions comprises defining a reachable simulation preorder relating the model and the tableau.

9. A method according to claim 7, wherein associating the transitions comprises finding a tableau transition that is not implemented in the model.

10. A method according to claim 9, wherein finding the tableau transition that is not implemented in the model comprises deriving an indication, based on the unimplemented transition, that the specification is not complete with respect to the model.

11. A method according to claim 9, wherein finding the tableau transition that is not implemented in the model comprises deriving an indication, based on the unimplemented transition, that a transition of the target system is missing in the model.

12. A method according to claim 1, and comprising associating model states with corresponding tableau states.

13. A method according to claim 12, wherein associating the model states with the corresponding tableau states comprises finding a tableau state that is not implemented in the model.

14. A method according to claim 13, wherein finding the tableau state that is not implemented in the model comprises deriving an indication, based on the unimplemented state, that the specification is not complete with respect to the model.

15. A method according to claim 13, wherein finding the tableau state that is not implemented in the model comprises deriving an indication, based on the unimplemented state, that a state of the target system is missing in the model.

16. A method according to claim 12, wherein associating the model states with the corresponding tableau states comprises finding multiple model states corresponding to a single tableau state.

17. A method according to claim 1, wherein creating the tableau comprises creating a reduced tableau from which one or more redundant states have been eliminated.

18. A method according to claim 1, wherein comparing the transitions comprises verifying that the specification is a complete and correct description of the implementation model responsive to the comparison.

19. A verification processor, which is configured to receive an implementation model, defining model states of a target system and model transitions between the model states, and to receive a specification of the target system, including properties that the system is expected to obey, and which is operative to create a tableau from the specification, the tableau defining tableau states with tableau transitions between the tableau states in accordance with the properties, and to compare the tableau transitions to the model transitions to determine whether a discrepancy exists therebetween.

20. A processor according to claim 19, which is operative to perform model checking of the implementation model.

21. A computer software product for verification of a specification of a target system, which specification includes properties that the system is expected to obey, by comparison with an implementation model, which defines model states of the target system and model transitions between the model states, the product comprising a computer-readable medium having computer program instructions recorded therein, which instructions, when read by a computer, cause the computer to create a tableau from the specification, the tableau defining tableau states with tableau transitions between the tableau states in accordance with the properties, and to compare the tableau transitions to the model transitions to determine whether a discrepancy exists therebetween.

22. A product according to claim 21, wherein the program instructions cause the computer to compare the tableau with the model by running a reachability analysis using both the implementation model and the tableau while providing identical inputs to the

inputs of both the implementation model and the tableau, and verifying that the outputs are always identical.

23. A product according to claim 22, wherein the reachability analysis is performed using an automatic model checker.

24. A product according to claim 21, wherein the instructions cause the computer to verify that the specification is a complete description of the implementation model.

25. A method for verification, comprising:

providing an implementation model, which defines model states of a target system and model transitions between the model states;

providing a specification of the target system, comprising properties that the system is expected to obey;

creating a tableau from the specification, the tableau defining tableau states with tableau transitions between the tableau states in accordance with the properties; and

comparing the model and the tableau by inputting the model and the tableau to an automatic model checking program.

26. A method according to claim 25, wherein creating the tableau comprises defining a finite state machine using a hardware description language.

27. A method according to claim 26, wherein the input model has model inputs and outputs, and wherein defining the finite state machine comprises describing a virtual device having inputs and outputs corresponding to the model inputs and outputs of the implementation model.

28. A method according to claim 27, wherein comparing the model and the tableau comprises running the model checker while providing identical inputs to the inputs of both the implementation model and the tableau, and verifying that the outputs are always identical.

29. A method according to claim 25, wherein comparing the model and the tableau comprises providing evidence of a transition or state in the tableau that is not implemented in the model.

30. A method according to claim 29, wherein providing the evidence comprises providing a counter-example indicative of the unimplemented transition or state.

31. Model checking apparatus, which is configured to receive an implementation model, defining model states of a target system and model transitions between the model states, and to receive a specification of the target system, including properties that the system is expected to obey, and which is operative to create a tableau from the specification, the tableau defining tableau states with tableau transitions between the tableau states in accordance with the properties, and to compare the tableau to the model by inputting the model and the tableau to an automatic model checking program.

32. A computer software product for verification of a specification of a target system, which specification includes properties that the system is expected to obey, by comparison with an implementation model, which defines model states of the target system and model transitions between the model states, the product comprising a computer-readable medium having computer program instructions recorded therein, which instructions, when read by a computer, cause the computer to create a tableau from the specification, the tableau defining tableau states with tableau transitions between the tableau states in accordance with the properties, and to compare the tableau to the model by inputting the model and the tableau to an automatic model checking program.

Evidence Appendix

NONE

Related Proceedings Appendix

NONE